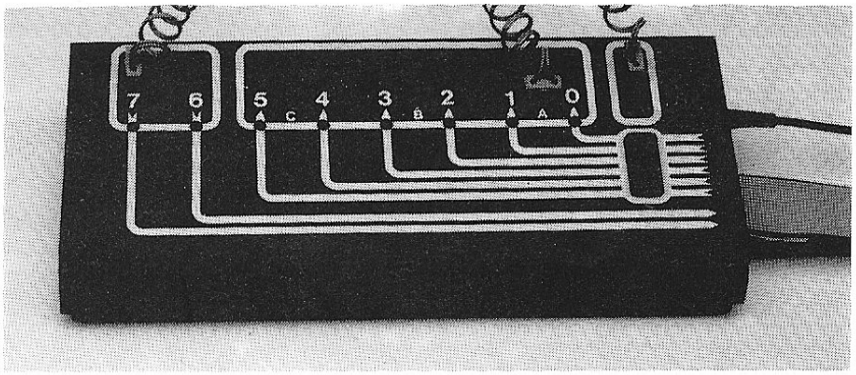
Beschreibung des LEGO Interface A



Kurzbeschreibung:

Das LEGO Interface A ist als Schul- und Ausbildungsinterface für die Prozeßsteuerung und die Robotik entwickelt worden.

Einsatzschwerpunkte sind die Informationstechnische Grundbildung, die Informationstechnik innerhalb des Technikunterrichts und die Informatik. Das Interface ist ein Teil der Lernpakete LEGO Technic Control, welche zusätzlich LEGO Systemsets und LEGO Programmierunterstützungen enthalten. Sie stellen Komplettlösungen für den Unterricht dar. Darüber hinaus kann das Interface separat benutzt werden. Zum übrigen LEGO System ist es kompatibel. Interface und Anschlußkabel zum Computer werden fertig montiert geliefert.

Technische Daten und Eigenschaften:

Daten:
2 Digitaleingänge
6 Schaltausgänge
1 Festspannungsausgang

Ausgangsspannung: 4 V Gleichspannung
Gesamte maximale Ausgangsleistung: 4 W

Die Spannungsversorgung des Interfaces erfolgt über ein *beigelegtes*, leistungsfähiges Netzteil.

Kontrollampen:
Die Zustände der Digitaleingänge werden durch grüne Kontrollampen, die der Ausgänge durch rote Kontrollampen angezeigt. Die rote Kontrollampe des Festspannungsausganges dient gleichzeitig als Bereitschaftsanzeige des Interfaces. Die Kontrollampen sind jeweils direkt unter der zugehörigen Anschlußbuchse angebracht.

Schaltausgänge:
Die sechs Schaltausgänge sind mit schnellschaltenden Leistungstransistoren ausgestattet und können als Ausgänge für Motoren, Lampen u. a. benutzt werden. *Sie lassen sich einzeln schalten.*
Separat verfügt das Interface über drei Motorenausgänge zur Beeinflussung von Drehrichtungen. Jeweils ein Paar der Schaltausgänge kann über eine zusätzliche Buchse auf dem Interface als umschaltbarer Motorenausgang benutzt werden.

Kabelverbindungen:
Kabelverbindungen zu den Modellen können mit den LEGO Anschlußkabeln *schnell und problemlos* hergestellt und geändert werden, ohne daß ein Schraubenzieher oder ein anderes Werkzeug nötig ist.

Sicherheit:
Das LEGO Interface ist *kurzschlußfest und verpolsicher*. Es verträgt beliebige Fehlschaltungen und kann auch "mal herunterfallen". Bei Anschlußaktionen zu den Modellen können weder das Interface noch der Computer noch weitere LEGO Systemkomponenten zu Schaden kommen. Die Datenübertragung zwischen Computer und Interface erfolgt optisch, d. h. über Optokoppler spannungsmäßig getrennt. Das Interface verfügt über einen *Notstop-Schalter*, der alle Schaltausgänge gleichzeitig unterbricht und eine komplexe Schalterkonstruktion in den Zuleitungen zu den Modellmotoren überflüssig macht.

Sprachfreundlichkeit:
Die Datenübertragung ist parallel. Hierdurch läßt sich das Interface im Vergleich zu einer seriellen Datenübertragung unvergleichlich viel einfacher von beliebigen Programmiersprachen ansprechen. Weitere Vorteile einer parallelen Datenübertragung sind höhere Übertragungsgeschwindigkeit und höhere Betriebssicherheit.

Didaktische Eigenschaften:

Kreativer Freiraum:
Besonders schulgeeignet ist das Interface durch seine technische Robustheit wie Kurzschlußfestigkeit und Verpolsicherheit (s. o.) *sowie durch die Möglichkeit, Modellanschlüsse auch bei einem eingeschalteten Computer und bei aktiver Software beliebig herzustellen und zu ändern. Der Lehrer braucht nicht ständig die Anschlußaktionen der Schüler zu überwachen.*

Didaktische Transparenz:
Sehr wichtig sind der *übersichtliche Aufdruck* auf dem stabilen Spezialgehäuse des Interfaces und die *sinnvoll angeordneten, verschiedenfarbigen Kontrollampen* mit dem Ziel einer *größtmöglichen Transparenz* für Schüler. Hierdurch sind alle Voraussetzungen dafür gegeben, daß Schüler *selbständig* herausfinden können, ob ein Steuerungsfehler seine Ursache beispielsweise im Modell, in einer falschen Anweisung oder in einer falschen Programmstruktur hat. Diese Fehlersuche wird durch umfangreiche Testhilfen der Software LEGO Lines noch weiter unterstützt.

Brücke zur Handsteuerung:
Die drei umschaltbaren Motorenausgänge des Interfaces entsprechen den drei Tastenpaaren der LEGO Steuerkonsole für die Handsteuerung. Hierdurch wird ein klarer didaktischer Übergang von der Handsteuerung zur Computersteuerung ermöglicht, wobei wesentliche Unterschiede nicht durch Anschlußprobleme verdeckt werden.

Handbuch:
Dem Interface ist ein *Handbuch beigelegt*. Das Handbuch gibt u. a. detaillierte Hinweise zur einzelnen Kontaktbelegung des Interfaceanschlusses zum Computer.

Für das LEGO Interface A sind Anschlüsse für die folgenden Computer erhältlich:

- Comodore 64, 128 - Anschluß an den USER-Port

- Apple IIe, II GS - Anschluß an einen Steckplatz

- IBM PC und kompatible Computer - Anschluß an einen Steckplatz

  Mehrere Interface gleichzeitig:
  Die LEGO Steckkarte enthält eine auftrennbare Kontaktbrücke, so daß eine andere Ansprechadresse gewählt werden kann. Hierdurch lassen sich zwei Steckkarte betreiben und damit zwei Interface gleichzeitig anschließen und steuern.
  Steckkartenprinzip:
  Durch die Verwendung des Steckkartenprinzips bleiben genormte Schnittstellen wie z. B. die Centronics-Schnittstelle weiterhin frei und für wichtige Peripherie wie den Drucker und andere nutzbar.

Die Artikel zu den Anschlußkabeln *enthalten auch die zugehörige Steckkarte und ein ausführliches Handbuch.*

Zukunftssicherheit:
Durch das LEGO Kabelkonzept mit Anschlußkabeln für verschiedene Computertypen bei gleichem Interface *sind die Unterrichtsmaterialien und die von Lehrern entwickelten Unterrichtskonzepte zukunftssicher.*

Softwareanbindung:

Sprachoffenheit:
Grundsätzlich kann das LEGO Interface durch eine *beliebige Software* angesprochen werden,
die Zugang zu Input/Output- Routinen hat.

Beispiele: - Commodore 64, 128: Peek, Poke Befehle in BASIC - IBM PC und kompatible Computer: INP, OUT
Detaillierte Hinweise liefern die Softwarehandbücher der Hersteller.

Dadurch ist das LEGO Interface und die Steuerung von LEGO Modellen *offen* gegenüber einem Ansprechen mit verschiedenen Computersprachen.

In dem LEGO Handbuch zum Anschlußkabel des Interface an den Commodore 64, 128 werden Grundprogramme in COMAL und BASIC vorgestellt. Entsprechendes gilt für den IBM PC und kompatible Computer in BASIC.

Programmierunterstützungen:
Besondere Hilfen bieten die LEGO Programmierunterstützungen LEGO Lines für die Computer Commodore 64, 128 und IBM PC, kompatible Computer sowie der LEGO TC Controller zu Turbo Pascal 4.0, 5.0 für den IBM PC und kompatible Computer.
Ergänzung: Update Diskette für den LEGO TC Controller zu Turbo Pascal 5.5 und Quick Pascal 1.0.

Preise:

In Abhängigkeit von Klassensatzbestellungen ergibt sich ein Preisbereich für das Interface A komplett mit Netzteil, Anschlußkabel, Steckkarte und Handbüchern von ca. DM 315,– bis ca. DM 390,–; Stand 1990.

Bezugsquelle:

Weitere Informationen:
LEGO GmbH, Schulbereich, Postfach 1263, 2354 Hohenwestedt

Beispielprogramme für die Ansteuerung des LEGO TC Interface in verschiedenen Computersprachen oder Benutzeroberflächen:

Die vorgestellten Beispielprogramme geben grundsätzliche Informationen über die problemlose Ansteuerung des LEGO Interface mit verschiedenen Computersprachen oder Benutzeroberflächen.

Die Beispielprogramme zeigen Lösungen mit Einsatz von Kontrollstrukturen für ein leicht verständliches Problem. Alle Ausgänge werden angesteuert und die Eingänge in verschiedenen Kombinationen abgefragt.

Die Modellsituation zeigt eine ampelgesteuerte Kreuzung mit einer Haupt- und einer Nebenstraße mit Bedarfssensor. Über einen Nachtschalter kann für die Nebenstraße gelbes Blinklicht eingeschaltet werden.

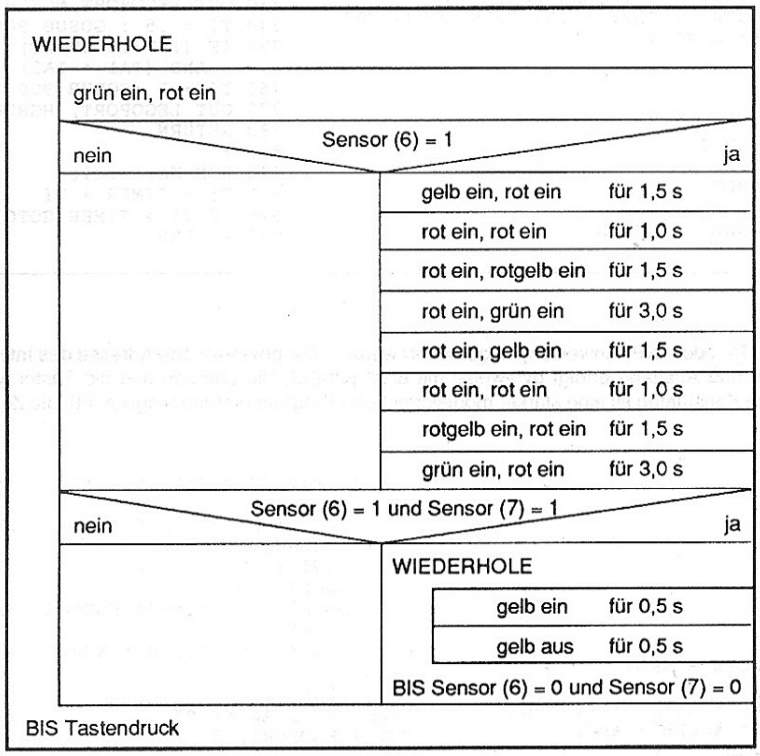
Anschlüsse am Interface:

0 : Nebenstraße grün	1 : Nebenstraße gelb	2 : Nebenstraße rot	3 : Hauptstraße grün
4 : Hauptstraße gelb	5 : Hauptstraße rot	6 : Bedarfssensor	7 : Nachtsensor

Ein Steuerprogramm, das auf die Tag-, Nachtsituation und die Bedarfsmeldung aus der Nebenstraße eingehen kann, soll geschrieben werden.

Der Grundzustand der Ampelanlage ist: Hauptstraße grün, Nebenstaße rot, Sensorwerte aus.

Programmstruktur:



---

1. LEGO Lines

Beim Start von LEGO Lines wird das Interface automatisch initialisiert.

Die Programmierung wird durch folgende Merkmale unterstützt:
- einfachste Eingabe
- deutsche Anweisungen und Umlaute
- Programmiermöglichkeit ohne Variablen

```
LEGO Lines                  7 6 5 4 3 2 1 0
                            ---------------
WIEDERHOLE                  . . . . . . . .
 grün          rot          * * 0 0 1 1 0 0
 WENN                       * 1 . . . . . .
  gelb         rot          * * 0 1 0 1 0 0   1,5
  rot          rot          * * 1 0 0 1 0 0   1,0
  rot          rotgelb      * * 1 0 0 1 1 0   1,5
  rot          grün         * * 1 0 0 0 0 1   3,0
  rot          gelb         * * 1 0 0 0 1 0   1,5
  rot          rot          * * 1 0 0 1 0 0   1,0
  rotgelb      rot          * * 1 1 0 1 0 0   1,5
  grün         rot          * * 0 0 1 1 0 0.  3,0
 ENDE WENN                  . . . . . . . .
 WENN                       1 0 . . . . . .
  WIEDERHOLE                . . . . . . . .
               gelb         * * 0 0 0 0 1 0   0,5
               aus          * * 0 0 0 0 0 0   0,5
  BIS                       0 0 . . . . . .
 ENDE WENN                  . . . . . . . .
IMMER                       . . . . . . . .
```

---

2. GWBASIC

Das Interface kann mit Hilfe der 'OUT'- oder 'INP'-Anweisung angesteuert werden. Die physikalische Adresse des Interface (hexadezimal 39D bzw. dezimal 925) ist bekannt. Die Ein- und Ausgabe erfolgt byteweise mit 8 Bit parallel. Die Lampen und die Taster werden über ihren Stellenwert angesprochen. Mit der Definition von Konstanten und der Unterprogrammtechnik (GOSUB Zeile) sind modellorientierte Programmabschnitte möglich.

```
100 REM Konstanten
110 :LEGOPORT = &H39D
120 REM Hauptstraße
130 :HROT = 32
140 :HGELB = 16
150 :HGRUEN = 8
160 REM Nebenstraße
170 :NROT = 4
180 :NGELB = 2
190 :NGRUEN = 1
200 REM Sensoren
210 :TA1 = 64
220 :TA2 = 128
230 :
400 REM Haupprogramm
410 OUT LEGOPORT,HGRUEN + NROT
420 IF (INP(LEGOPORT) AND TA1) = TA1 THEN GOSUB 500
430 IF (INP(LEGOPORT) AND (TA1 + TA2) = TA2 THEN GOSUB 700
440 A$ = INKEY$: IF A$ = "" GOTO 420
450 OUT LEGOPORT,0
460 END
470 :
500 REM Bedarfsampel
510 OUT LEGOPORT, HGELB + NROT
520 TI = 1.5 :GOSUB 900
530 OUT LEGOPORT, HROT + NROT
540 TI = 1 :GOSUB 900
550 OUT LEGOPORT, HROT  + NROT + NGELB
560 TI = 1.5 :GOSUB 900
570 OUT LEGOPORT, HROT + NGRUEN
580 TI = 3 :GOSUB 900
590 OUT LEGOPORT, HROT + NGELB
600 TI = 1.5 :GOSUB 900
610 OUT LEGOPORT, HROT + NROT
620 TI = 1 :GOSUB 900
630 OUT LEGOPORT, HROT + HGELB + NROT
640 TI = 1.5 :GOSUB 900
650 OUT LEGOPORT, HGRUEN + NROT
660 TI = 3 :GOSUB 900
670 RETURN
680 :
700 REM Nachtbetrieb
710 OUT LEGOPORT,0
720 TI = 1 : GOSUB 900
730 OUT LEGOPORT,NGELB
740 TI = .5 : GOSUB 900
750 IF (INP(LEGOPORT)
      AND (TA1 + TA2) <> 0 GOTO 710
760 TI = 1 :GOSUB 900
770 OUT LEGOPORT, HGRUEN + NROT
780 RETURN
810 :
900 REM Wartezeit
910 TI = TIMER + TI
920 IF TI > TIMER GOTO 920
930 RETURN
```

---

3. Turbo Basic, Version A

Das Interface kann mit Hilfe der 'OUT'- oder 'INP'-Anweisung angesteuert werden. Die physikalische Adresse des Interface (hexadezimal 39D bzw. dezimal 925) ist bekannt. Die Ein- und Ausgabe erfolgt byteweise mit 8 Bit parallel. Die Lampen und die Taster werden über ihren Stellenwert angesprochen. Mit der Definition von Konstanten ist eine stärker modellorientierte Programmierung möglich. Für die Zeitsteuerung kann die 'DELAY'-Anweisung verwendet werden.

```
REM Konstanten
    %LEGOPORT = &H39D
    %GRUEN = 1
    %GELB = 2
    %ROT  = 4
    %TA1 = 64
    %TA2 = 128

REM Programm
    OUT %LEGOPORT, 8 * %GRUEN + %ROT
    DO
     IF (INP(%LEGOPORT) AND %TA1) = %TA1 THEN
         OUT %LEGOPORT, 8 * %GELB + %ROT
         DELAY(1.5)
         OUT %LEGOPORT, 8 * %ROT + %ROT
         DELAY(1 )
         OUT %LEGOPORT, 8 * %ROT + %ROT+%GELB
         DELAY(1.5 )
         OUT %LEGOPORT, 8 * %ROT + %GRUEN
         DELAY(3 )
         OUT %LEGOPORT, 8 * %ROT + %GELB
         DELAY(1.5 )
         OUT %LEGOPORT, 8 * %ROT + %ROT
         DELAY(1 )
         OUT %LEGOPORT, 8 * (%ROT+%GELB) + %ROT
         DELAY(1.5 )
         OUT %LEGOPORT, 8 * %GRUEN + %ROT
         DELAY(3 )
     END IF
     IF (INP(%LEGOPORT) AND (%TA1 + %TA2)) = %TA2 THEN
         DO
          OUT %LEGOPORT, 8 * 0 + %GELB
          DELAY(.5)
          OUT %LEGOPORT, 0
          DELAY(.5)
         LOOP UNTIL (INP(%LEGOPORT) AND (%TA1 + %TA2)) = 0
         DELAY(.5)
         OUT %LEGOPORT, 8 * %GRUEN + %ROT
     END IF
     A$ = INKEY$
    LOOP UNTIL A$ <> ""
    OUT %LEGOPORT, 0
    END
```

Turbo Basic, Version B

Die Unterprogrammtechnik (CALL Name)
ermöglicht übersichtliche Programme.

```
REM Konstanten
    %LEGOPORT = &H39D
    %GRUEN = 1
    %GELB = 2
    %ROT  = 4
    %TA1 = 64
    %TA2 = 128

REM Haupprogramm
    CALL LICHT(%GRUEN, %ROT)
    DO
     IF (INP(%LEGOPORT) AND %TA1) = %TA1 THEN
      CALL Bedarfsampel
     END IF
     IF (INP(%LEGOPORT) AND (%TA1 + %TA2)) = %TA2 THEN
      CALL Nachtbetrieb

     END IF
     A$ = INKEY$
    LOOP UNTIL A$ <> ""
    CALL LICHT(0,0)
```

```
SUB Bedarfsampel
    CALL LICHT(%GELB, %ROT)
    DELAY(1.5)
    CALL LICHT(%ROT, %ROT)
    DELAY(1 )
    CALL LICHT(%ROT, %ROT+%GELB)
    DELAY(1.5 )
    CALL LICHT(%ROT, %GRUEN)
    DELAY(3 )
    CALL LICHT(%ROT, %GELB)
    DELAY(1.5 )
    CALL LICHT(%ROT, %ROT)
    DELAY(1 )
    CALL LICHT(%ROT+%GELB, %ROT)
    DELAY(1.5 )
    CALL LICHT(%GRUEN, %ROT)
    DELAY(3 )
END SUB
```

```
SUB Nachtbetrieb
    DO
     CALL LICHT(0, %GELB)
     DELAY(.5)
     CALL LICHT(0,0)
     DELAY(.5)
    LOOP UNTIL (INP(%LEGOPORT) AND (%TA1 + %TA2)) = 0
    DELAY(.5)
    CALL LICHT(%GRUEN, %ROT)
END SUB

SUB LICHT(H,N)
    OUT %LEGOPORT,H * 8 + N
END SUB
```

---

4. Quick Basic

Das Interface kann mit Hilfe der 'OUT'- oder 'INP'-Anweisung angesteuert werden. Die physikalische Adresse des Interface (hexadezimal 39D bzw. dezimal 925) ist bekannt. Die Ein- und Ausgabe erfolgt byteweise mit 8 Bit parallel. Die Lampen und die Taster werden über ihren Stellenwert angesprochen.

Mit der Definition von Konstanten ist eine stärker modellorientierte Programmierung möglich. Die Unterprogrammtechnik (GOSUB Name) ermöglicht übersichtliche Programme. Die Wartezeit in der Zeitsteuerung muß über den 'TIMER'-Wert berechnet werden.

```
REM Konstanten
    LEGOPORT = &H39D
REM Hauptstraße
    HROT = 32
    HGELB = 16
    HGRUEN = 8
REM Nebenstraße
    NROT = 4
    NGELB = 2
    NGRUEN = 1
REM Sensoren
    TA1 = 64
    TA2 = 128

REM Hauptprogramm
    OUT LEGOPORT, HGRUEN + NROT
    A$ = INKEY$
    WHILE A$ = ""
     IF (INP(LEGOPORT) AND TA1) = TA1
     THEN GOSUB Bedarfsampel
     IF (INP(LEGOPORT) AND (TA1 + TA2)) = TA2
     THEN GOSUB Nachtbetrieb
     A$ = INKEY$
    WEND
    OUT LEGOPORT,0
    END
```

```
Bedarfsampel:
    OUT LEGOPORT, HGELB + NROT
    TI = 1.5 :GOSUB Wartezeit
    OUT LEGOPORT, HROT + NROT
    TI = 1 :GOSUB Wartezeit
    OUT LEGOPORT, HROT  + NROT + NGELB
    TI = 1.5 :GOSUB Wartezeit
    OUT LEGOPORT, HROT + NGRUEN
    TI = 3 :GOSUB Wartezeit
    OUT LEGOPORT, HROT + NGELB
    TI = 1.5 :GOSUB Wartezeit
    OUT LEGOPORT, HROT + NROT
    TI = 1 :GOSUB Wartezeit
    OUT LEGOPORT, HROT + HGELB + NROT
    TI = 1.5 :GOSUB Wartezeit
    OUT LEGOPORT, HGRUEN + NROT
    TI = 3 :GOSUB Wartezeit
    RETURN

Nachtbetrieb:
    NACHT = (INP(LEGOPORT) AND (TA1 + TA2))
    WHILE NACHT <> 0
     OUT LEGOPORT,NGELB
     TI = .5 : GOSUB Wartezeit
     OUT LEGOPORT,0
     TI = .5 : GOSUB Wartezeit
     NACHT = (INP(LEGOPORT) AND (TA1 + TA2))
    WEND
    TI = 1 :GOSUB Wartezeit
    OUT LEGOPORT, HGRUEN + NROT
    RETURN

Wartezeit:
    TI = TIMER + TI
Loop:
    IF TI > TIMER GOTO Loop
    RETURN
```

---

5. Comal

Das Interface kann mit Hilfe der Comal-Package 'System' über die 'OUT'- und 'INP'-Anweisung angesprochen werden. Die physikalische Adresse des Interface (hexadezimal 39D bzw. dezimal 925) ist bekannt. Die Ein- und Ausgabe erfolgt byteweise mit 8 Bit parallel. Die Lampen und die Taster werden über ihren Stellenwert angesteuert. Die Wartezeit in der Zeitsteuerung muß über den 'TIMER'-Wert berechnet werden.
Mit Hilfe von hardwareorientierten Prozeduren und Konstanten läßt sich eine modellorientierte Programmierumgebung schaffen.

```
0010 //Ampelsteuerung
0020 USE system
0030 gruen:=1
0040 gelb:=2
0050 rot:=4
0060 rotgelb:=6
0070 ampelaus
0080 hauptstrasse(gruen)
0090 nebenstrasse(rot)
0100 //
0110 REPEAT
0120   x$:=KEY$
0130   IF taste_gedrückt=1 THEN bedarfsampel
0140   IF taste_gedrückt=2 THEN nachtbetrieb
0150 UNTIL x$<>""
0160 END
0170 //
0180 PROC bedarfsampel
0190   hauptstrasse(gelb)
0200   warte(1.5)
```

```
0210   hauptstrasse(rot)
0220   warte(1)
0230   nebenstrasse(rotgelb)
0240   warte(1.5)
0250   nebenstrasse(gruen)
0260   warte(3)
0270   nebenstrasse(gelb)
0280   warte(1.5)
0290   nebenstrasse(rot)
0300   warte(1)
0310   hauptstrasse(rotgelb)
0320   warte(1.5)
0330   hauptstrasse(gruen)
0340   warte(3)
0350 ENDPROC bedarfsampel
0360 //
0370 PROC nachtbetrieb
0380   ampelaus
0390   WHILE (taste_gedrückt<>0) DO
0400     nebenstrasse(gelb)
```

```
0410      warte(1)
0420      ampelaus
0430      warte(1)
0440    ENDWHILE
0450    warte(1)
0460    nebenstrasse(rot)
0470    hauptstrasse(gruen)
0480 ENDPROC nachtbetrieb
0490 //
0500 PROC ampelaus
0510    out(925,0)
0520 ENDPROC ampelaus
0530 //
0540 PROC hauptstrasse(nr)
0550    out(925,nr*8+(inp#(925) BITAND 7))
0560 ENDPROC hauptstrasse
0570 //
0580 PROC nebenstrasse(nr)
0590    out(925,nr+(inp#(925) BITAND 56))
0600 ENDPROC nebenstrasse
0610 //
0620 FUNC taste_gedrückt
0630    CASE (inp#(925) BITAND 192) OF
0640    WHEN 0
0650      RETURN 0
0660    WHEN 128
0670      RETURN 2
0680    OTHERWISE
0690      RETURN 1
0700    ENDCASE
0710 ENDFUNC taste_gedrückt
0720 //
0730 PROC warte(n) CLOSED
0740    ti:=TIMER+n
0750    REPEAT UNTIL ti<=TIMER
0760 ENDPROC warte
```

---

6. Turbo Pascal und Quick Pascal

Das Interface kann mit Hilfe des vordefinierten Arrays Port angesteuert werden. Port ist als array of Byte deklariert und erlaubt direkte Zugriffe auf die Ein/Ausgabeadressen des Systems. Die physikalische Adresse des Interface (hexadezimal 39D bzw. dezimal 925) ist bekannt. Die Ein- und Ausgabe erfolgt byteweise mit 8 Bit parallel. Die Lampen und die Taster werden über ihren Stellenwert angesprochen.

Mit der Definition von Konstanten ist eine stärker modellorientierte Programmierung möglich.
Für die Zeitsteuerung kann die 'DELAY'-Anweisung verwendet werden.

```
Program Ampelanlage_1;
uses Crt;
const
    LEGOPORT = $39D;
    GRUEN    = 1;
    GELB     = 2;
    ROT      = 4;
    TA1    = 64;
    TA2    = 128;

procedure Licht(H,N:byte);
 begin
  port[LEGOPORT] := H * 8 + N;
 end;

procedure Warte(zeit:real);
 begin
  delay(round(1000*zeit));
 end;

procedure Bedarfsampel;
 begin
    Licht(GELB, ROT);
    warte(1.5);
     Licht(ROT, ROT);
    warte(1 );
     Licht(ROT, ROT+GELB);
    warte(1.5 );
     Licht(ROT, GRUEN);
    warte(3 );
     Licht(ROT, GELB);
    warte(1.5 );
     Licht(ROT, ROT);
    warte(1 );
     Licht(ROT+GELB, ROT);
    warte(1.5 );
     Licht(GRUEN, ROT);
    warte(3 );
end;

procedure Nachtbetrieb;
 begin
    repeat
      Licht(0,0);
     warte(0.5);
      Licht(0, GELB);
     warte(0.5);
    until (port[LEGOPORT] AND (TA1 + TA2)) = 0;
    warte(0.5);
     Licht(GRUEN, ROT);
end;

{ Hauptprogramm }
begin
    Licht(GRUEN, ROT);
    repeat

     if (port[LEGOPORT] AND TA1) = TA1 THEN  Bedarfsampel;
     if (port[LEGOPORT] AND TA2) = TA2 THEN  Nachtbetrieb;
    until keypressed;
     Licht(0,0);
end.
```

---

7. Turbo Pascal mit LEGO TC Controller / Quick Pascal mit LEGO TC Controller

Durch den Aufruf des Controllers wird das Interface automatisch initialisiert und eine Direktsteuerung bereitgestellt. Die Unit LTCC stellt einen interfacespezifischen Befehlssatz bereit: z.B. Sensor(x) oder BitOn(x). Die Anschlüsse können einzeln, d.h. bitweise angesprochen werden. Die Unit AMP ist ein Beispiel für einen problemspezifischen Befehlssatz: z.B. Hauptstrasse(gruen) oder warte(1).

```
Program Ampelanlage_2;

uses Crt, LTCC, AMP;

begin
 repeat
  Hauptstrasse(gruen); Nebenstrasse(rot);
  If Sensor(6) = 1 then
  begin
   Hauptstrasse(gelb); warte(1.5);
   Hauptstrasse(rot); warte(1);
   Nebenstrasse(rotgelb); warte(1.5);
   Nebenstrasse(gruen); warte(3);
   Nebenstrasse(gelb); warte(1.5);
   Nebenstrasse(rot); warte(1);
   Hauptstrasse(rotgelb);warte(1.5);
   Hauptstrasse(gruen);warte(3);
  end; Unit Amp;

 If (Sensor(6) = 0) and (Sensor(7) = 1) then
  begin
   Hauptstrasse(aus);
   repeat
    Nebenstrasse(gelb); warte(0.5);
    Nebenstrasse(aus);  warte(0.5);
   until (Sensor(6) = 0) and (Sensor (7) = 0);
  end;
 until keypressed;
end.
```

```
Unit Amp;
{ ------------------------------ }
   { Anschlüsse am Interface:
   ------------
   0:   gruen     _
   1:   gelb      __   Nebenstrasse
   2:   rot       _
   3:   gruen     _
   4:   gelb      __   Hauptstrasse
   5:   rot       _
   6:   Sensor für Bedarfsampel
   7:   Sensor für Nachtbetrieb
   ------------------------------ }

 interface
 { --- }
 uses DOS,CRT,LTCC;

  type farben = (rot,rotgelb,gruen,gelb,aus);

   procedure Hauptstrasse(farbe:farben);
   procedure Nebenstrasse(farbe:farben);
   procedure Warte(zeit:real);
{------------------------------ }
```

```
implementation
{ ------ }
Procedure Hauptstrasse(farbe:farben);
 begin
  case farbe of
   rot      : begin BitOff(3);BitOff(4);BitOn(5); end;
   rotgelb  : begin BitOff(3);BitOn(4); BitOn(5); end;
   gruen    : begin BitOn(3); BitOff(4);BitOff(5); end;

   gelb     : begin BitOff(3);BitOn(4); BitOff(5); end;
   aus      : begin BitOff(3);BitOff(4);BitOff(5); end;
  end;
 end;

Procedure Nebenstrasse(farbe:farben);
 begin
  case farbe of
   rot      : begin BitOff(0);BitOff(1);BitOn(2); end;
   rotgelb  : begin BitOff(0);BitOn(1);BitOn(2); end;
   gruen    : begin BitOn(0);BitOff(1);BitOff(2); end;
   gelb     : begin BitOff(0);BitOn(1);BitOff(2); end;
   aus      : begin BitOff(0);BitOff(1);BitOff(2); end;
  end;
 end;

 Procedure Warte(zeit:real);
 begin wait(zeit) end;

begin
 Resetcontroller;
end.
{ ------------------------------ }
```

Durch die Verwendung der Units läßt sich das Programm klar gliedern und problemspezifisch formulieren. Selbstverständlich kann auf die Verwendung einer Unit verzichtet werden. Dann wird die Programmformulierung hardwarespezifischer.

---

8. LogSim

Der Aufbau einer Ampelsteuerung mit logischen Gattern führt schnell zu einer komplexen Schaltung. Über einen 4-Bit-Zähler werden die Interface-Ausgänge angesteuert. Die hexadezimale Adresse des Interface und die Bit-Nummer werden in das Ausgabebausteinsymbol eingetragen. Für die Ampelphasen wurde die folgende Wahrheitstabelle realisiert:

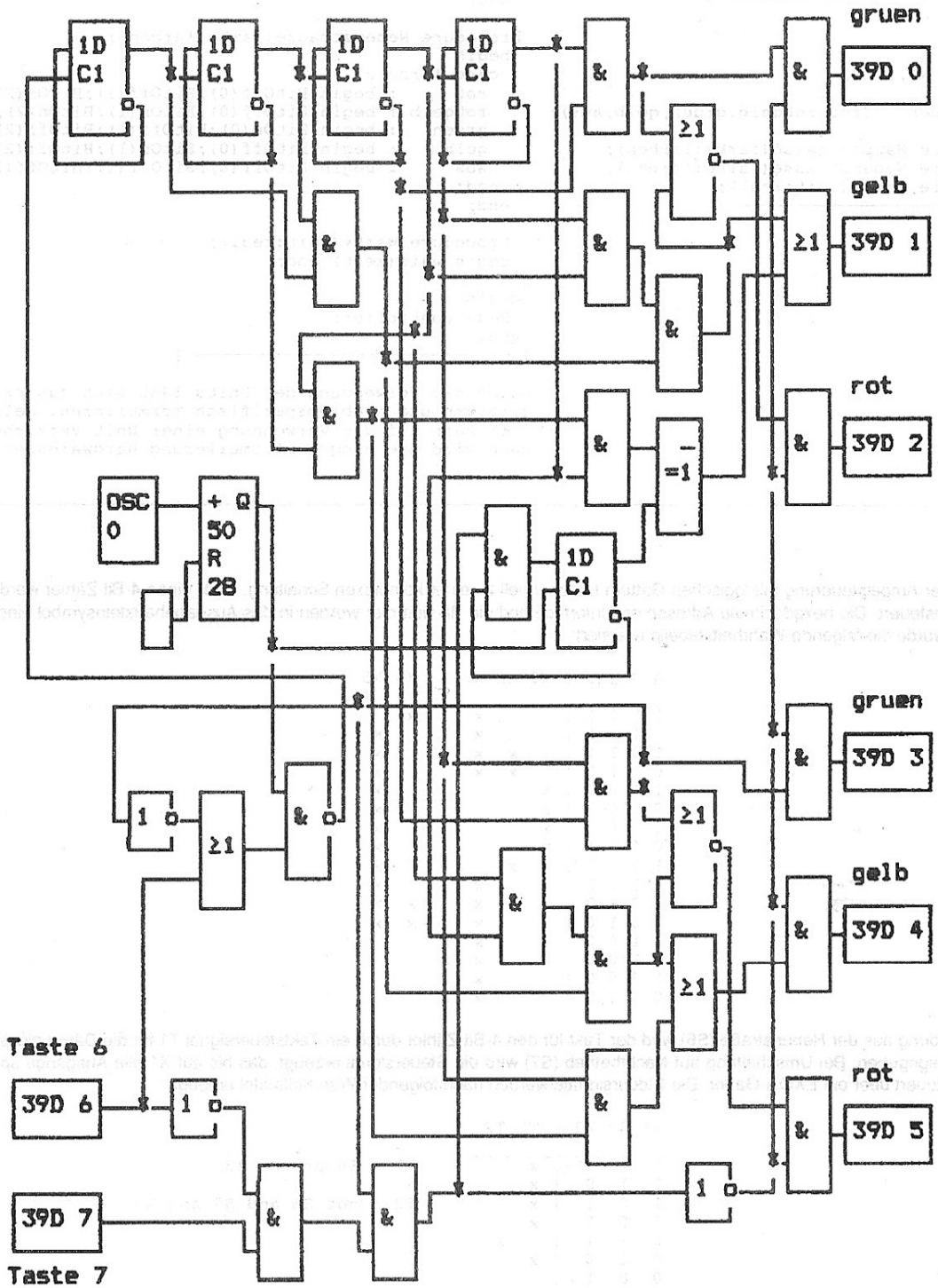
A	B	C	D	X0	X1	X2	X3	X4	X5
1	1	1	1			x		x	
0	1	1	1			x			x
1	0	1	1		x	x			x
0	0	1	1		x	x			x
1	1	0	1	x					x
0	1	0	1	x					x
1	0	0	1	x					x
0	0	0	1	x					x
1	1	1	0		x				x
0	1	1	0			x			x
1	0	1	0			x		x	x
0	0	1	0			x		x	x
1	1	0	0			x	x		
0	1	0	0			x	x		
1	0	0	0			x	x		
0	0	0	0			x	x		

Bei Bedarfsmeldung aus der Nebenstraße (S6) wird der Takt für den 4-Bit-Zähler durch ein Taktsteuersignal T1 für die Dauer eines Ampelzyklus freigegeben. Bei Umschaltung auf Nachtbetrieb (S7) wird ein Steuersignal erzeugt, das bis auf X1 alle Ausgänge sperrt. X1 blinkt im Ampeltakt, gesteuert über ein EXOR-Gatter. Die Steuersignale werden nach folgender Wahrheitstafel erzeugt:

S6	S7	X3	T1	T2
1	1	1	x	
1	1	0	x	
1	0	1	x	
1	0	0	x	
0	1	1		x
0	1	0	x	
0	0	1		
0	0	0	x	

T1 = S6 or not X3

T2 = not S6 and S7 and X3

Ampelsteuerung über das LEGO Interface mit einem 4-Bit-Zähler

Bedarfsanforderung über Eingang 6
Umschaltung auf Nachtbetrieb mit Eingang 7